Multilingwis – A Multilingual Search Tool for Multi-Word Units in Multiparallel Corpora

Clematide, Simon ; Graën, Johannes ; Volk, Martin

Abstract: We describe a web-based application for searching translations of multi-word units in large, openly available multiparallel corpora. This web application offers a unique resource for multilingual terminologists and translators. The first edition of the tool covers the debates of the European Parliament in five languages: English, French, German, Italian, and Spanish. Our search tool provides a simple and intuitive user interface, which optimally supports content-oriented queries while relieving the user from specifying complicated search expressions in a complex query language. We describe the necessary automatic preprocessing steps of the linguistic data, the retrieval component, and the techniques needed for offering a zero configuration search.

Multilingwis – A Multilingual Search Tool for Multi-word Units in Multiparallel Corpora

Simon Clematide
Institute of Computational Linguistics
University of Zurich
siclemat@cl.uzh.ch

Johannes Graën
Institute of Computational Linguistics
University of Zurich
graen@cl.uzh.ch

Martin Volk
Institute of Computational Linguistics
University of Zurich
volk@cl.uzh.ch

Keywords: Multilinguality, Linguistic Search, Parallel Documents, Translation, Dictionary

Abstract

We describe a web-based application for searching translations of multi-word units in large, openly available multiparallel corpora. This web application offers a unique resource for multilingual terminologists and translators. The first edition of the tool covers the debates of the European Parliament in five languages: English, French, German, Italian, and Spanish. Our search tool provides a simple and intuitive user interface, which optimally supports content-oriented queries while relieving the user from specifying complicated search expressions in a complex query language. We describe the necessary automatic preprocessing steps of the linguistic data, the retrieval component, and the techniques needed for offering a zero configuration search.

1. INTRODUCTION

Large collections of multiparallel texts, i.e. multilingual documents with aligned paragraphs or sentences across all languages, are openly available. For instance, debates from the European parliament (Koehn 2005), administrative and legislative texts from the EU (Steinberger et al. 2012; Hajlaoui et al. 2014), official documents of the UN in 6 languages (Eisele & Chen 2010) as well as translated movie subtitles (Tiedemann 2012). These corpora are highly useful and valuable for translators, terminologists, and contrastive corpus linguists if they can be exploited effectively.

Multilingwis[1] is a web-based search tool for multiparallel word-aligned corpora that allows its users to find translations[2] of multi-word units efficiently and easily in any of the available languages. The tool is optimized for quick ad-hoc searches and explorations of translation variants and supports content-oriented access to translated multi-word units across multiple languages (typically complex noun phrases, but clearly searches for single words are also supported). Our goal is to relieve the user from specifying complex search expressions in a corpus query language, and we put substantial effort in providing a zero-configuration query interface that just works as expected. As shown in Figure 1, a user can enter "las violaciones de los derechos

[1] https://pub.cl.uzh.ch/purl/multilingwis

[2] Throughout this paper we use the term *translation* to refer to words that express the same content in parallel texts. Finding a translation via a search system as *Multilingwis* does not imply that the search hits are direct translations or that the search words were written in their original language.

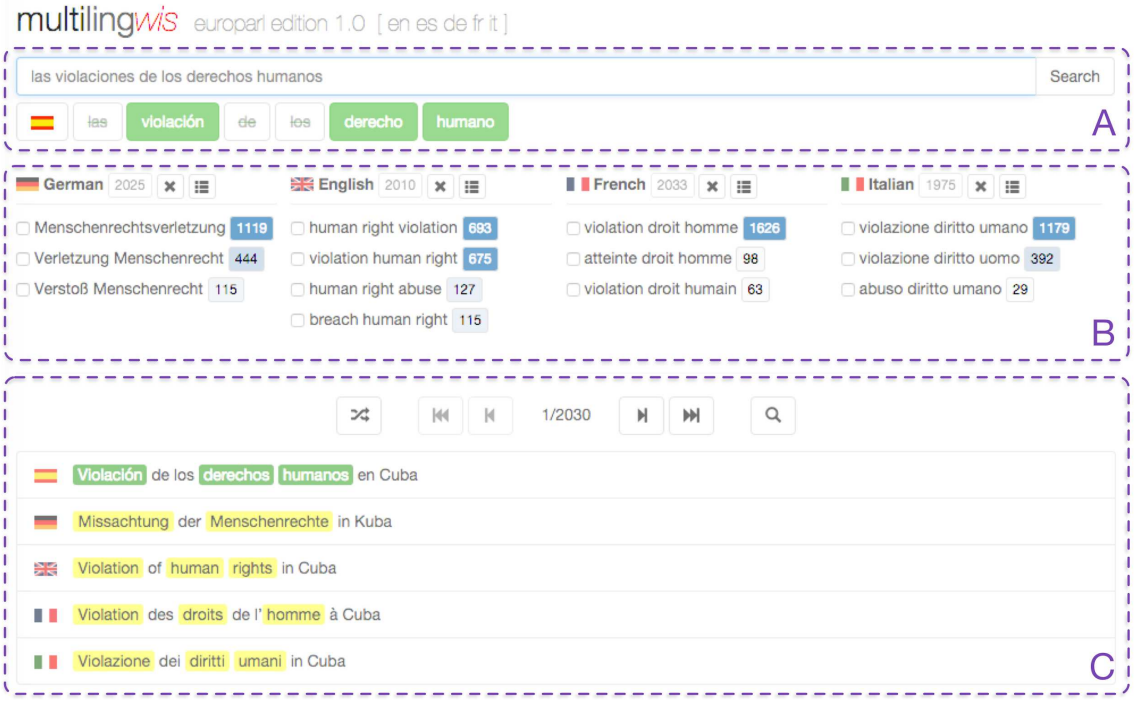

Figure 1: User interface with interaction zones A, B, and C marked up

humanos" and the system automatically recognizes the language and reduces the input to a sequence of the following lemmatized content words "violación derecho humano", which then will be searched in the corpus. While the order of the search terms must match the order in the sentences of the search language, there is, of course, no order restriction in the corresponding parallel sentences, e.g. we find the following lemmatized English translation variants "human right violation" and "violation human right", which are most popular. As the search is restricted to content words, any parts of speech other than nouns, verbs, adjectives, and adverbs are ignored. In the example sentences presented to the user, naturally, the inflected form including the function words will be shown, e.g. "human rights violations" or "violations of human rights".

Multilingwis is one outcome of an interdisciplinary research project[3] in corpus linguistics and computational linguistics which aims at an automatically computed rich and multi-layered annotation of multiparallel corpora, including the automatic alignment of text units (e.g. speech turns in debates), sentences, and words. In addition to these levels of alignment, we aim at the sub-sentential alignment of noun groups. In order to fully exploit such complex annotated linguistic structures, a full-fledged linguistic query language is needed with an accordingly steep learning curve for the user. In contrast to a complex query language, *Multilingwis* offers an easy-to-use access to our annotated and aligned data.

1.1. Related Work

There are a number of web-based search systems available for finding translations of words in parallel corpora. The main benefit of such tools lies the fact that the user immediately sees real-world usage examples of translation pairs in the context of sentences, thus the user is able to judge whether the translation is adequate for a given domain or register. *Linguee*[4] currently supports bilingual searches for 25 languages.

[3] http://www.cl.uzh.ch/research/parallelcorpora/sparcling_en.html
[4] http://www.linguee.de

Systems such as the highly multilingual translation sharing platform *TAUS DATA*[5], the so-called 'bilingual concordancer' *TradooIT*[6], or *Bilingwis*[7] offer similar functionality for bilingual searches. See Volk et al. (2014) for a more detailed discussion about their usability, the covered languages, and the amount of integrated parallel data.

We know of only two systems which offer searches in multiparallel corpora. The *OPUS-Corpus Query* system[8] (Tiedemann 2012) allows to query several large multiparallel corpora of the OPUS collection using the efficient query infrastructure of the *Corpus Workbench* (CWB) (Evert & Hardie 2011). However, the presentation of the search results is not good because it just shows the parallel sentences and does not include any highlighting of the word-aligned translations of the user's search words. The user must therefore read through the parallel sentences and spot the potential translations by himself. *ParaSol*[9] (von Waldenfels 2011) started as a specialized parallel corpus collection for many Slavic languages with 1 to 4 millions of tokens per language. Additionally, it contains now texts in Romance, Germanic, Baltic, and other European languages. However, due to licencing issues of the text material it is restricted to academic research purposes. *ParaSol* also uses the CWB as its linguistic query engine and also lacks a highlighting of the word translations in the parallel sentences as in the OPUS-Corpus query system.

Our goal is to provide a user-friendly experience of multilingual translation spotting, especially, highlighting the aligned translations in the example sentences and providing frequency distributions of translation patterns which allow the user to quickly identify the most prominent translations variants in order.

2. PREPROCESSING OF THE LINGUISTIC DATA

We extracted parallel text units in English, French, German, Italian and Spanish from the *Corrected & Structured Europarl Corpus*[10] (Graën et al. 2014), to each of which we subsequently applied the TreeTagger (Schmid 1994) for tokenization, part-of-speech (PoS) tagging and lemmatization. Tagging was done with the language models available from the TreeTagger's web page[11]. We adapted the TreeTagger's tokenizer (abbreviation lexicons, punctuation) and extended its tagging lexicon (especially the German one) with lemmas and PoS tags for frequent words unknown to the language models.

Table 1 shows the amount of tokens per language and the fraction of distinct word forms (=types) which received a TreeTagger lemma. In total, we count 220 million tokens comprising 1 million distinct word forms out of which 214,585 lemmas have been identified by our adapted TreeTagger pipeline. The differences between languages are substantial: the Spanish language model has a low rate of properly lemmatized words, whereas German has the highest absolute number of lemmatized words but due to its large number of distinct word forms cannot reach the lemmatization rate of English. If a token did not receive a TreeTagger lemma, we default it to the word form.

[5] http://www.tausdata.org
[6] http://www.tradooit.com
[7] http://pub.cl.uzh.ch/purl/bilingwis
[8] http://opus.lingfil.uu.se/bin/opuscqp.pl
[9] http://www.parasolcorpus.org
[10] Altogether 146,652 speech turns are available in all these five languages in CoStEP, which bases on Europarl release v7 (Koehn 2005) and can be obtained freely from http://www.statmt.org/europarl/.
[11] http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/#Linux

Language	*Tokens*	*Types*	*TT Lemmas*	*TT Fraction*
English	43m	127,105	73,250	57.6%
French	47m	142,898	83,937	58.7%
German	41m	367,159	174,885	47.6%
Italian	43m	181,478	108,147	59.6%
Spanish	45m	175.817	75,187	42.8%

Table 1: Distribution of tokens, types (=distinct word forms), TreeTagger (TT) lemmas, and the fraction of types which received a TreeTagger lemma

We assigned universal part-of-speech tags to each token using the mapping for language-specific tagsets defined by Petrov et al. (2012), and added a few more mappings for some model-specific tags of the TreeTagger. Universal part-of-speech tags helped us to easily separate content words from function words across all languages.[12] Each language has about 22 million content words in our data set.

For sentence alignment, a refined sentence splitting was necessary because some languages use colons or semicolons where others prefer a full stop. For example, the English sentence "However, we have also been guided by another factor, namely the lack of progress on the question of an energy tax." is separated by a colon in French ("Toujours est-il qu'il existait également un autre facteur: l'absence de progrès en matière d'imposition fiscale.") and a full stop in Spanish ("No obstante, había otra cuestión. La del estancamiento en el tema del impuesto energético."). Refined sentence boundaries were identified by language-specific rules based on part-of-speech tags and lemmas. Pairwise bilingual sentence alignments was then carried out by the statistical sentence aligner *hunalign* (Varga et al. 2005). Each language has about 1.7 million sentences, with a total of about 16 million pairwise sentence alignments.

For word alignment, we applied the standard tool *GIZA++* (Och & Ney 2003) for each language pair and each direction, resulting in 20 sets of directed 1:n alignments of content words (only adjectives, adverbs, nouns and verbs were aligned). We symmetrized these sets by constructing the union of alignments (see Tiedemann 2011, p.76), thus favoring recall for our application. A total of 110 million content words was the basis for our word alignment.

The linguistic data obtained (tokens with lemmas and part-of-speech tags, sentence segments with their pairwise alignments and word alignments for content words for each language pair) is stored in a relational database. Database features such as *multi-column indexes*, *materialized views* and *stored procedures* allow for an efficient search and retrieval of the corpus data.

[12] As content words, we select adjectives (ADJ), adverbs (ADV), nouns (NOUN), verbs (VERB) (including auxiliary verbs). All other 8 categories are treated as function words (including prepositions). Some fine-tuning of this simple classification could improve the results.

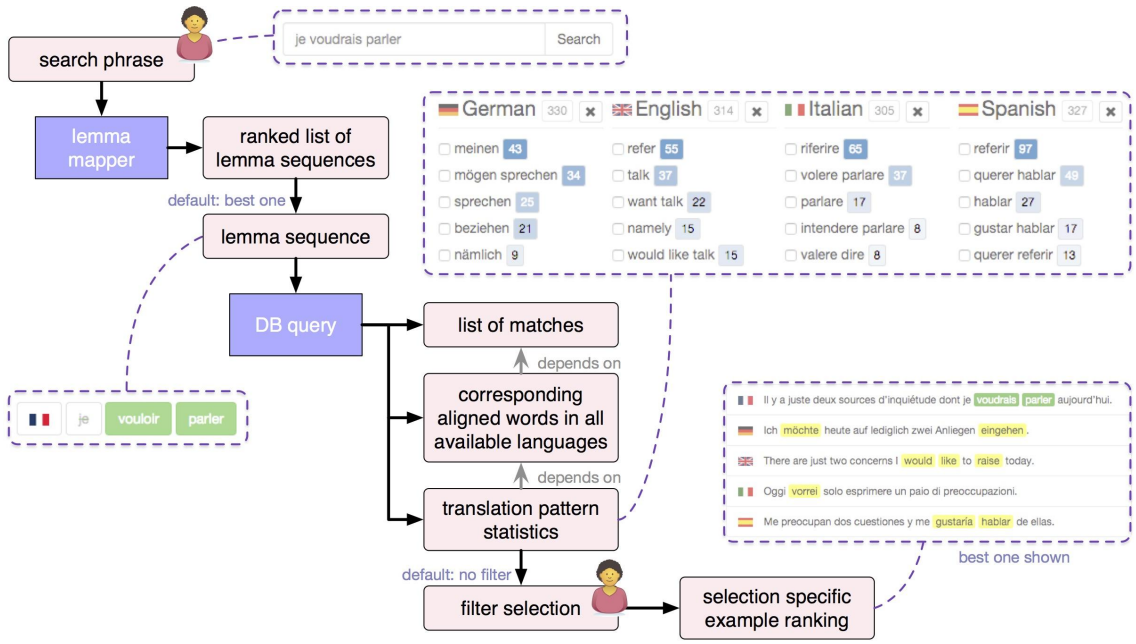

Figure 2: Overall system architecture of *Multilingwis*

3. SYSTEM IMPLEMENTATION

3.1. User Interface

Our user-friendly search interface allows the user to enter a stretch of text in a single input field and to immediately get back a reasonable result set, much alike typical web search interfaces work nowadays. This kind of user-friendliness requires that our application automatically transforms the search string into an appropriate query for the retrieval component. It involves the following steps: (a) tokenization of the input (e.g. separating clitic articles from nouns as in French, "l'auto" splits into "l'" and "auto"), (b) automatic language identification (limited to the 5 languages currently present in our system), (c) removal of function words (e.g. articles or prepositions, as our system only deals with content words), and (d) lemmatization of inflected content word forms (the lemmas are taken directly from the TreeTagger output). Thus, an input as "las violaciones de los derechos humanos" will be reduced to the following sequence of Spanish content lemmas: "violación derecho humano". We ensure that the system reduces the search strings into the most probable sequence of lemmas of the most probable language. Due to vocabulary overlap between languages, code-switching and proper names, quite a number of word forms can be found in several languages. We also deal with the case that the user enters a sequence of already perfectly lemmatized content words and make sure that each lemma is mapped to itself in the transducer using the frequency counts of its most frequent word form.

If the automatic language identification does not work as expected, the user can select a language-specific search. In this search mode, we also present a drop-down selection for each word form that can be analyzed into more than one lemma. For instance, the inflected word "accepted" can either be a verb with the lemma "accept", or an adjective with the lemma "accepted". We will also develop an advanced search form which supports the search for inflected surface forms and filtering for certain parts of speech, e.g. the lemma "break" can be a noun or an adjective in English, which have different translations.

As can be seen in Figure 1, we present the results of a query in *Multilingwis* in 3 zones. The first zone A shows the user how his input has been interpreted, that is, the sequence of content lemmas searched by the retrieval component. If there is at least one hit, the second zone B displays the distribution of all corresponding translation patterns. At the same time, the third zone C renders an example sentence with the highlighted translations. All example sentences are ordered by a score for good examples (GDEX score), which in our case is currently a value that favors short sentences with a small deviation of sentence length across all languages. More sophisticated GDEX scores as mentioned in the literature (Rychlý et al. 2008) could be computed offline and stored as an attribute for each sentence.

Zone B contains for each language the distribution of all translation variants, in descending order of frequency ("human right violation, human right abuse, breach of human rights"). Each translation variant has a check box attached to it, and the user can enforce the display of an example sentence showing this very translation by checking it. Therefore, zone B offers the user a flexible facility for faceted search refinement. Additionally, each translation variant in zone B provides a hyperlink directed to a new search. Every translation variant can again be queried by a single click, and therefore supports quick explorations across different cross-lingual or monolingual verbalizations of the same concepts.

3.2. Retrieval Component

The first step in the retrieval transforms the query input string into a language-specific sequence of lemmatized content words. This step is implemented by finite-state technology (Lindén et al. 2013), which also allows us to efficiently normalize orthographic variants.

For the language-unspecific search mode, we basically need a transducer that encodes for each language a mapping for each (inflected) word form into its most frequent lemma. Each lemma is annotated by its language code and its frequency class[13] (the higher the better) in order to guess the language of the user query string. For instance, the string "die ganze EU" (*the whole EU*) results in the following decorated lemmas for German, "die/de/19 ganz/de/11 EU/de/14", and for English, "die/en/7 EU/en/15", thus preferring German by simply summing up the frequency classes. Function words such as articles are important for language identification, but their lemmas are ignored for the search. According to our experience, such word-based language identification works efficiently, and we see no need to use an external language identifier based on character n-grams (e.g. Lui & Baldwin (2012)), especially, given the fact that such language identifiers typically need at least 30-60 characters for a precise language prediction.

For the language-specific search mode, we built a similar transducer for each language that encodes a mapping for each word form into all admissible lemmas. Each lemma is decorated by its frequency class in order to present ambiguous lemmatizations in descending order of frequency. For instance, *accepted* occurs 4,535 times with the lemma *accept* and only 97 times with the lemma *accepted*, therefore, the default lemmatization of *accepted* is *accept*.

[13] We start from the formula for logarithmic frequency classes which are independent of the corpus size and compare the frequency of a word w against the frequency of the most frequent word w_{max}: $N(w)$ = floor(0.5-log2(freq(w)/freq(w_{max})). If a word is in frequency class N, it means that the most frequent word is N times more frequent than w. We transform the numeric value N according to N'(w)=abs(N(w)-N(1/w_{max})-1) in order to implement language identification as a maximization of the sum of all N'(w) of a language; for unseen words we set N'=0.

We cannot expect that our users type the search words exactly as we store them in our database. Therefore, we allow a range of spelling variants, for instance, accented or special characters such as “ß” in German or “ç” in French are optionally mapped to their ASCII representation “ss” resp. “c”. Furthermore, we handle spelling variants concerning hyphens. If a user searches for “proeuropäischen” (‘pro-European’), he is offered both lemmas, “pro-europäisch” and “proeuropäisch”.

Once the language-specific sequence of lemmatized content words has been derived from the user input, a database function takes over the tasks to (a) search for a matching sequence of tokens at intervals of at most 4 tokens where the interjacent tokens can only be function words, (b) look up word alignments for each token of each matching sequence[14], and (c) build a statistics of translation patterns on top of it. Figure 2 depicts these steps in the context of the user interface.

Step (a), the search for matching token sequences, starts with a lookup of the first search lemma in a database index based on lemmas and token positions. For every following search lemma, the result is subsequently intersected with another lookup in that index which is limited to the next 4 positions of the previous token found. The tokens in between the matching ones are subsequently filtered for not containing any content word, i.e. their universal part-of-speech tag not being a verb, noun, adjective or adverb.

In step (b), the token sequences are intersected with a database index on the symmetrical word alignments, such that the result set comprises a list of matching tokens in the source language, a list of corresponding tokens in each target language for which we have word alignments, and a sequence of lemmas (=translation variant).

Step (c) ranks these translation variants according to their frequency for each language. The ranking is displayed in zone B.

The whole data set of tokens in the source and target languages together with the respective translation variants are kept until the user performs a new search (either by entering a new query or by clicking on one of the translation variant buttons). Whenever a translation variant is selected or deselected, the sentences shown in zone C get updated with the supposedly best example that matches the intersection of any checked translation variant between all languages. If none is chosen, which is the default configuration, all translation variants of the particular language are considered. If there is no example translation for the current selection, the user is advised accordingly.

4. DISCUSSION AND FUTURE WORK

Every preprocessing step for our corpus data can be improved further: Our corpus still contains some misaligned text units. Subsequent sentence and word alignment cannot work for these cases since alignment depend on correct alignment on the text level. In sentence pairs where we align non-corresponding text, our statistical word alignment tool GIZA++ will nonetheless return the most probable alignments, which results in a long tail of incorrect translation variants that occur only once. Therefore, we currently work on the detection of sentences which are not parallel.

Given the fact that we align related languages where translated words often have a similar shape on the level of characters (so-called *cognates*), a more informed approach than the one used by GIZA++ could produce better results (see Sojka et al. (2012)).

For the current version, we already provide domain-adapted external lexicons for the TreeTagger with PoS tags and lemmas, however, there is still room for improved lemmatization. A lot of the more administrative and technical terms in Europarl are not

[14] We call the corresponding sequence of lemmas a translation variant.

covered by the current TreeTagger models. Another open issue are ambiguous lemmas in the TreeTagger output, for instance, the Italian word "sono" is analyzed into "essere|sonare", which represents the two admissible alternative lemmatizations. We currently store the unmodified TreeTagger lemmas in our database. However, this distorts the translation statistics for the verb "essere". We should therefore either try to disambiguate the ambiguous lemmas (e.g. by preferring the globally more frequent lemma), or we should implement a proper Boolean search for such cases.

Another improvement concerning lemmatization is related to German verbs with separable prefixes, e.g. "ansprechen" (*to address, to speak about*). If such verbs are used as finite forms in main clauses, the finite verb and its prefix are in different topological fields. For instance, "Wir **sprechen** die wichtigen Probleme nicht **an**" (*we do not address the important problems*). In order to provide a proper overview of the translations of "ansprechen", we should attach the prefix "an" to the verb lemma "sprechen" in such cases. This can be done quite reliably and is already implemented by the aforementioned system *Bilingwis* (Volk et al. 2011).

A further question concerning lemmatization is the treatment of words with numbers, e.g. "62jährig" (*62 years old*) in German. Currently, the user has to enter the exact number in order to find translations, which is a bit cumbersome. What a typical user probably would like to see are translation patterns of "DDjährig" where "DD" could be any sequence of digits.

Our formula for the GDEX score currently only considers the consistent shortness of sentences across languages. Although frequent translation patterns will be shown more often than rare ones for obvious reasons, we plan to integrate the frequency of translation patterns into the ranking of the examples.

A different *Multilingwis* edition, for instance, one based on the United Nations corpus with Arabic, Chinese, English, French, Russian, and Spanish would connect less related languages in a single view. A *Multilingwis* edition with movie subtitles could even be interesting for language learners, however, the text quality (OCR errors, spelling errors) will need some attention.

We did our best to provide an intuitive and practical user interface. In order to gain a better understanding whether our design decisions were adequate, we need to perform usability tests with users interested in multilingual texts and observe via eye tracking devices how they actually interact with our web interface while performing some tasks with *Multilingwis*.

Acknowledgements

This research was supported by the Swiss National Science Foundation under grant 105215_146781/1 through the project "SPARCLING – Large-scale Annotation and Alignment of Parallel Corpora for the Investigation of Linguistic Variation".

References

Eisele, A. & Chen, Y., 2010. MultiUN: A Multilingual Corpus from United Nation Documents. In *Proc LREC 2010*. pp. 2868–2872.

Evert, S. & Hardie, A., 2011. Twenty-first century Corpus Workbench: Updating a query architecture for the new millennium. In *Proceedings of Corpus Linguistics 2011*.

Graën, J., Batinic, D. & Volk, M., 2014. Cleaning the Europarl Corpus for Linguistic Applications. In *Konvens 2014*. Stiftung Universität Hildesheim.

Hajlaoui, N. et al., 2014. DCEP - Digital Corpus of the European Parliament. In *Proc LREC 2014*. pp. 3164–3171.

Koehn, P., 2005. Europarl: A parallel corpus for statistical machine translation. In *Machine Translation Summit*. pp. 79–86.

Lindén, K. et al., 2013. Using HFST for Creating Computational Linguistic Applications. In *Computational Linguistics*. Springer Berlin Heidelberg, pp. 3–25.

Lui, M. & Baldwin, T., 2012. Langid.Py: An Off-the-shelf Language Identification Tool. In *Proceedings of the ACL 2012 System Demonstrations*. ACL '12. Stroudsburg, PA, USA: Association for Computational Linguistics, pp. 25–30.

Och, F.J. & Ney, H., 2003. A Systematic Comparison of Various Statistical Alignment Models. *Computational linguistics*, 29(1), pp.19–51.

Petrov, S., Das, D. & McDonald, R., 2012. A Universal Part-of-Speech Tagset. In *Proc LREC 2012*. pp. 2089–2096.

Rychlý, P. et al., 2008. GDEX: Automatically Finding Good Dictionary Examples in a Corpus. In *Proceedings of the XIII EURALEX International Congress*. pp. 425–432.

Schmid, H., 1994. Probabilistic Part-of-Speech Tagging Using Decision Trees. In *Proceedings of International Conference on New Methods in Language Processing*. pp. 44–49.

Sojka, P. et al., 2012. Text, Speech and Dialogue. In Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 370–377.

Steinberger, R. et al., 2012. DGT-TM: A freely available Translation Memory in 22 languages. In *Proc LREC 2012*. pp. 454–459.

Tiedemann, J., 2011. *Bitext Alignment*, Morgan & Claypool Publishers.

Tiedemann, J., 2012. Parallel Data, Tools and Interfaces in OPUS. In *Proc of LREC 2012*. pp. 2214–2218.

Varga, D. et al., 2005. Parallel corpora for medium density languages. *Proc RANLP*, pp.590–596.

Volk, M. et al., 2011. Word-aligned parallel text : a new resource for contrastive language studies. In *Supporting Digital Humanities, Conference 2011*.

Volk, M., Graën, J. & Callegaro, E., 2014. Innovations in Parallel Corpus Search Tools. In *Proc LREC 2014*. pp. 3172–3178.

Von Waldenfels, R., 2011. Recent developments in ParaSol: Breadth for depth and XSLT based web concordancing with CWB. In M. Daniela & R. Garabik, eds. *Natural Language Processing, Multilinguality. Proceedings of Slovko 2011*. Bratislava, pp. 156–162.